

# An Introduction to *FilterFFPE*

*Lanying Wei*

Modified: 20 August, 2020. Compiled: October 27, 2020

## Contents

1	Introduction . . . . .	1
2	Input . . . . .	1
3	Artifact Chimeric Read Filtration . . . . .	1
A	Session info . . . . .	3

## 1 Introduction

---

The next-generation sequencing (NGS) reads from formalin-fixed paraffin-embedded (FFPE) samples contain numerous artifact chimeric reads, which can lead to a large number of false positive structural variation (SV) calls. The *FilterFFPE* package finds and filters these artifact chimeric reads from BAM files of FFPE samples.

## 2 Input

---

The required input is an indexed BAM file of the FFPE sample, the PCR or optical duplicates should be marked or removed from the BAM file. Example of such a BAM file is stored in the 'extdata' directory of *FilterFFPE* package).

## 3 Artifact Chimeric Read Filtration

---

The filtration includes two steps: 1) Find artifact chimeric reads from BAM file . 2) Remove these artifact chimeric reads in the filtered BAM file. We recommend to also remove PCR or optical duplicates of all chimeric reads, since these reads may contain duplicates of artifact chimeric reads. `findArtifactChimericReads` can be used to find artifact chimeric reads, read names of PCR or optical duplicates of all chimeric reads are also found and written in a txt file by this function. `filterBamByReadNames` can be used for further filtration, it generates a filtered and indexed BAM file. `FFPESReadFilter` combines these two functions.

## An Introduction to *FilterFFPE*

```
> library(FilterFFPE)
> # Find artifact chimeric reads
> file <- system.file("extdata", "example.bam", package = "FilterFFPE")
> outFolder <- tempdir()
> FFPEReadsFile <- paste0(outFolder, "/example.FFPEReads.txt")
> dupChimFile <- paste0(outFolder, "/example.dupChim.txt")
> artifactReads <- findArtifactChimericReads(file = file, threads = 2,
+                                           FFPEReadsFile = FFPEReadsFile,
+                                           dupChimFile = dupChimFile)
> head(artifactReads)

[1] "SRR1523265.12888813" "SRR1523265.22338213"
[3] "SRR1523265.24545253" "SRR1523265.2726740"
[5] "SRR1523265.31420529" "SRR1523265.31521425"

>
```

```
> # Filter artifact chimeric reads and PCR or optical duplicates of chimeric reads
> dupChim <- readLines(dupChimFile)
> readsToFilter <- c(artifactReads, dupChim)
> destination <- paste0(outFolder, "/example.FilterFFPE.bam")
> filterBamByReadNames(file = file, readsToFilter = readsToFilter,
+                      destination = destination, overwrite=TRUE)

[1] "/tmp/RtmpEPnhFZ/example.FilterFFPE.bam"

>
```

```
> # Perform finding and filtering with one function
> file <- system.file("extdata", "example.bam", package = "FilterFFPE")
> outFolder <- tempdir()
> FFPEReadsFile <- paste0(outFolder, "/example.FFPEReads.txt")
> dupChimFile <- paste0(outFolder, "/example.dupChim.txt")
> destination <- paste0(outFolder, "/example.FilterFFPE.bam")
> FFPEReadFilter(file = file, threads=2, destination = destination,
+               overwrite=TRUE, FFPEReadsFile = FFPEReadsFile,
+               dupChimFile = dupChimFile)

[1] "/tmp/RtmpEPnhFZ/example.FilterFFPE.bam"

>
```

The generated BAM file can be loaded with `scanBam` function from *Rsamtools* package for further interrogation.

```
> # load Bam file with scanBAM
> newBam <- Rsamtools::scanBam(destination)
> head(newBam[[1]]$seq)

DNAStringSet object of length 6:
  width seq
[1] 90 CAGTGCTCAACCACTCCTCTCT...CCCTGGCCCTCCCAGCCCACGAT
[2] 90 CAGTGCTCAACCACTCCTCTCT...CCCTGGCCCTCCCAGCCCACGAT
[3] 90 CAGTGCTCAACCACTCCTCTCT...CCCTGGCCCTCCCAGCCCACGAT
[4] 90 CAGTGCTCAACCACTCCTCTCT...CCCTGGCCCTCCCAGCCCACGAT
```

```
[5] 90 ACCCCACTCCCTGGCCCTCCCAGC...CCTGAACCCCCAGCCTGTGGTTC  
[6] 90 CCCCCACTCCCTGGCCCTCCCAGC...CCTGAACCCCCAGCCTGTGGTTC
```

## A Session info

---

```
> packageDescription("FilterFFPE")  
  
Package: FilterFFPE  
Type: Package  
Title: FFPE Artificial Chimeric Read Filter for NGS  
      data  
Version: 1.0.0  
Authors@R: person("Lanying", "Wei",  
                  email="lanying.wei@uni-muenster.de", role =  
                  c("aut", "cre"), comment = c(ORCID =  
                  "0000-0002-4281-8017"))  
Description: This package finds and filters  
             artificial chimeric reads specifically  
             generated in next-generation sequencing (NGS)  
             process of formalin-fixed paraffin-embedded  
             (FFPE) tissues. These artificial chimeric reads  
             can lead to a large number of false positive  
             structural variation (SV) calls. The required  
             input is an indexed BAM file of a FFPE sample.  
License: LGPL-3  
Encoding: UTF-8  
Imports: foreach, doParallel, GenomicRanges, IRanges,  
         Rsamtools, parallel, S4Vectors  
Suggests: BiocStyle  
biocViews: StructuralVariation, Sequencing,  
           Alignment, QualityControl, Preprocessing  
git_url:  
  https://git.bioconductor.org/packages/FilterFFPE  
git_branch: RELEASE_3_12  
git_last_commit: 8838232  
git_last_commit_date: 2020-10-27  
Date/Publication: 2020-10-27  
Author: Lanying Wei [aut, cre]  
       (<https://orcid.org/0000-0002-4281-8017>)  
Maintainer: Lanying Wei <lanying.wei@uni-muenster.de>  
Built: R 4.0.3; ; 2020-10-28 00:08:40 UTC; unix  
  
-- File: /tmp/RtmphChwGj/Rinst49fc6be60e88/FilterFFPE/Meta/package.rds  
  
> sessionInfo()  
  
R version 4.0.3 (2020-10-10)  
Platform: x86_64-pc-linux-gnu (64-bit)  
Running under: Ubuntu 18.04.5 LTS
```

```
Matrix products: default
BLAS: /home/biocbuild/bbs-3.12-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.12-bioc/R/lib/libRlapack.so

locale:
 [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8      LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
 [9] LC_ADDRESS=C             LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
 [1] stats      graphics  grDevices  utils      datasets
 [6] methods    base

other attached packages:
 [1] FilterFFPE_1.0.0

loaded via a namespace (and not attached):
 [1] knitr_1.30           XVector_0.30.0
 [3] GenomicRanges_1.42.0 BiocGenerics_0.36.0
 [5] zlibbioc_1.36.0     IRanges_2.24.0
 [7] BiocParallel_1.24.0 doParallel_1.0.16
 [9] rlang_0.4.8         foreach_1.5.1
[11] GenomeInfoDb_1.26.0 tools_4.0.3
[13] parallel_4.0.3      xfun_0.18
[15] iterators_1.0.13    htmltools_0.5.0
[17] yaml_2.2.1          digest_0.6.27
[19] crayon_1.3.4        GenomeInfoDbData_1.2.4
[21] BiocManager_1.30.10 codetools_0.2-16
[23] S4Vectors_0.28.0    bitops_1.0-6
[25] RCurl_1.98-1.2      evaluate_0.14
[27] rmarkdown_2.5       compiler_4.0.3
[29] Rsamtools_2.6.0     Biostrings_2.58.0
[31] stats4_4.0.3        BiocStyle_2.18.0
```